REMARKS/ARGUMENTS

Claims 1-30 are pending in the application. Reconsideration in view the following remarks is respectfully requested.

Claims 21 and 27 are objected to under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Claims 4, 5, 6, 7, 8, 12, 15, 16, 20, 23, 24, 28,13, 14, 21, 22, 29, and 30 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Claims 1, 2, 3, 4, 7, 8, 9, 10, 11, 12, 15, 16, 17, 18, 19, 20, 23, 24, 25, 26, 27, and 28 are rejected under 35 U.S.C. 102(e) as being clearly anticipated by Sager, US Patent 6,542,921 ("Sager ('921)"). Claim 5, 13, 21 and 29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sager, WO 01/04750 A1 ("Sager ('750)"), cited as prior art reference in paper number 6, filed on November 8, 2002, in view of Jacobsen et al., US Patent 6, 606, 639 ("Jacobsen").

In this Amendment, claim 7 is amended to correct minor clerical errors. Claims 18, 19 and 27 are amended to overcome 35 U.S.C. 112 second paragraph objections to minor clerical errors.

In response to the 35 U.S.C. 112 second paragraph rejections of claims 4, 5, 6, 7, 8, 12, 15, 16, 20, 23, 24, 28, 13, 14, 21, 22, 29 and 30, claims 4, 8, 12, 16, 20, 24 and 28 have been amended to more distinctly claims the present invention.

With respect to the §§ 102(e) and 103(a) rejections, Applicants note that the Sager '921 names as its sole inventor, a co-inventor of the present application. Furthermore, it

is noted that the Sager ('750) reference is mostly the same disclosure. In discussing Sager ('750), Applicants make no representations that Sager ('750) was published prior to the invention date of the present invention.

Applicants respectfully submit nowhere in Sager ('921) is the disclosure, teaching or suggestion of "[i]n a processor to handle processing of at least first and second threads in parallel, a method of assigning thread priority comprising: assigning priority to said first thread; loading a preliminary value to a thread precedence counter; and assigning priority to said second thread after said thread precedence counter expires" (e.g., the embodiment as claimed in claim 1).

The Office Action asserts Sager has taught assigning priority to said first thread (abstract, Figure 9, element 909), loading a preliminary value to a thread precedence counter (abstract, Figure 9, element 905, "Priority Period"), and assigning priority to said second thread in response to expiration of said thread precedence counter (abstract, Figure 9, element 917). The Abstract states:

> The present invention provides a method and apparatus for controlling a processing priority assigned alternately to a first thread and a second thread in a multithreaded processor to prevent deadlock and livelock problems between the first thread and the second thread. In one embodiment, the processing priority is initially assigned to the first thread for a first duration. It is then determined whether the first duration has expired in a given processing cycle. If the first duration has expired, the processing priority is assigned to the second thread for a second duration.

Column 11 line 5 describes the operation of element 905 as follows –

> At block 905, the priority duration or priority period for each thread is initialized to some predetermined period. The priority duration is defined as a duration or interval of time during which a particular thread is given the processing priority. In one embodiment, the priority duration is measured in terms of processing periods where each processing period corresponds to a predetermined number of processing cycles and each processing cycle corresponds to a predetermined number of processor clock periods. In one embodiment, an initial priority duration for each thread is set to be equal to a predetermined number of clock periods, for example 16 medium clock periods.

Column 11 line 17 described the operation of element 909 as follows –

> At block 909, the processing priority is given to one of the two threads for a time period corresponding to its priority duration. In one embodiment, the thread precedence bit is set to a first value, for example 0, to indicate that the processing priority is given to thread 0 and set to a second value, for example 1, to indicate that the processing priority is given to thread 1. In one embodiment, the priority duration for a particular thread starts running as soon as the processing priority is assigned to that particular thread.

Column 11 line 25 describes the operation of element 917 as follows –

> At decision loop 913, the method 900 proceeds to block 917 if the current priority duration has expired. At block 917, the processing priority is alternated, i.e., assigned to the other thread.

The Sager ('921) reference discloses that during the thread priority processing process, the determinations is made whether a first *duration* has expired in a given processing cycle and if so, the processing priority is assigned to the second thread for a second *duration*. Again, in element 905, Sager discloses that it is then determined whether the first *duration* has expired in a given processing cycle and if the first duration has expired, the processing priority is assigned to the second thread for a second duration. The Sager reference further describes similar "processing period[s]" and "priority duration[s]".

However, Applicant respectfully submits that nowhere in Sager (including the Figures and cited sections) is the disclosure, teaching or suggestion of at least "a method of assigning thread priority comprising...*loading a preliminary value to a thread precedence counter*" as recited in the embodiment of amended claim 1. Further support can be found in page 6 line 8 through page 7 line 14 of the specification. Since the Sager patent does not include the disclosure, teaching or suggestion of at least a *thread precedence counter* or the *loading a preliminary value into such a thread precedence counter*, the 102(e) rejection is lacking and should be withdrawn. Independent claim 25 contains substantively similar elements and should be allowed for similar reasons.

Claims 2-6 and 26-30 depend from allowable independent claims 1 and 25 respectively, and therefore should be allowed as well.

Furthermore, Applicants' respectfully submit that nowhere is the disclosure, teaching or suggestion of "...[i]n a processor to handle processing of at least first and second threads in parallel, a method of assigning thread priority comprising: assigning priority to said first thread; and assigning priority to said second thread when one of a pluarlity of conditions is true, the conditions including: whether processing of said first thread retires an instruction from said first thread; and whether and there is not an indication of approaching instruction side starvation for said first thread" (e.g., the embodiment as claimed in amended claim 7).

The Office Action asserts that Sager has taught "...a multi-threaded processor...if there is not an indication of approaching instruction side starvation for said first thread (Figure 9, elements 913 and 917, When the priority of the second thread is assigned in response to only the current priority period expiring, element 913, the condition is true that there is not an indication of approaching instruction side starvation for said thread.)"

As discussed above, the cited sections in Sager disclose that based upon a determination whether the first duration has expired in a given processing cycle has expired, the processing priority is assigned to the second thread for a second duration. However, Applicant respectfully submits that nowhere in Sager (including the Figures and cited sections) is the disclosure, teaching or suggestion of at least "... assigning priority to said second thread when one of a plurality of conditions is true, the conditions including: whether processing of said first thread retires an instruction from said first thread; *and whether there is not an indication of approaching instruction side starvation*

for said first thread". Further support for the indication of a approaching instruction side starvation can be found on page 9 line 3 of the specification. Applicants respectfully submit that since each and every element is not taught, suggested or disclosed by the cited reference, the 102(e) rejection is lacking and should be withdrawn. Independent claims 15 and 23 contain substantively similar claim elements, and therefore should be allowed as well. Dependent claims 8, 16 and 24 depend from allowable base claims and therefore should be allowed as well.

Additionally, Applicants' respectfully submit that nowhere is the disclosure, teaching or suggestion of "...[a] computer system to handle processing of at least first and second threads in parallel, comprising: a memory to store instructions for first and second threads; a processor including control logic coupled to said memory to assign priority between said first and second threads; a thread precedence counter coupled to said control logic wherein priority is assigned to said second thread after said thread precedence counter expires" (e.g., the embodiment as claimed in claim 17).

The Office Action asserts that Sager discloses a thread precedence counter coupled to said control logic wherein priority is assigned to a said second thread after said thread precedence counter expires (abstract, Figure 9, element 905, element 917, "Priority Period"). However, Applicants submit that nowhere in Sager (including the cited sections and Figures) is the teaching, suggestion or disclosure of a "...thread precedence counter coupled to said control logic wherein priority is assigned to said second thread after said thread precedence counter expires". Support for the thread precedence counter can be found a page 6 line 18. However, such thread precedence is not taught, suggested or disclosed anywhere in Sager. Since each and every claim

element is not taught, suggested or disclosed by the cited reference, the 102(e) rejection is lacking and should be withdrawn. Independent claims 1, 9, 17 and 25 contain substantively similar claim elements and therefore should be allowed as well. Claims 2-6, 10-14, 18-22 and 26-30 depend from the aforementioned allowable independent claims, and therefore are in condition for allowance as well.

With regard to the 35 U.S.C. 103(a) rejections of claims 5, 6, 13, 14, 21, 22, 29 and 30, Applicants respectfully these claims depend from allowable independent claims and therefore are allowable as well.

For at least all the above reasons, the Applicants respectfully submit that this application is in condition for allowance. A Notice of Allowance is earnestly solicited.

The Examiner is invited to contact the undersigned at (408) 975-7500 to discuss any matter concerning this application. The Office is hereby authorized to charge any additional fees or credit any overpayments under 37 C.F.R. § 1.16 or § 1.17 to Deposit Account No. **11-0600**.

Respectfully submitted,
KENYON & KENYON

Dated: July 12, 2004 By: _____
Sumit Bhattacharya
(Reg. No. 51,469)
Attorneys for Intel Corporation

KENYON & KENYON
333 W. San Carlos St., Suite 600
San Jose, CA 95110
Telephone: (408) 975-7500
Facsimile: (408) 975-7501